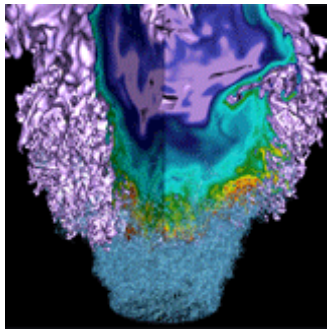
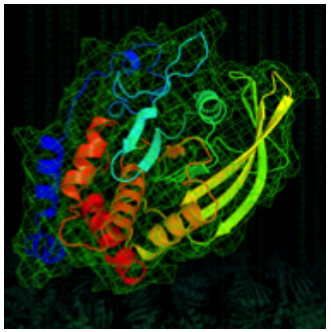
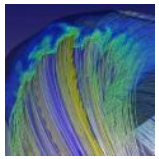
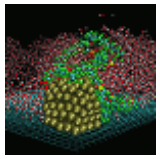
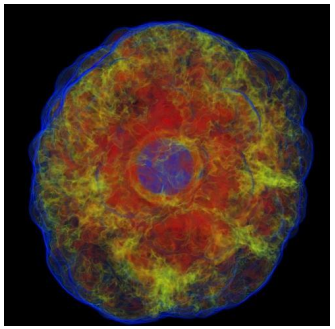
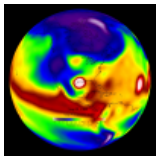
NUG Monthly Telecon


NERSC 40 YEARS at the FOREFRONT 1974-2014

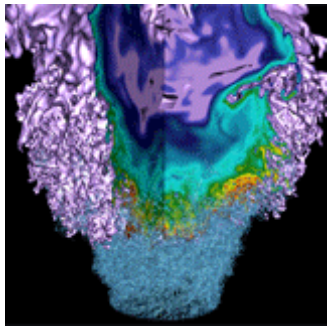
October 8th 2015

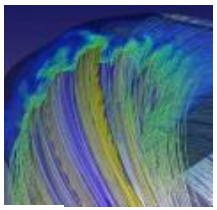
U.S. DEPARTMENT OF ENERGY | Office of Science

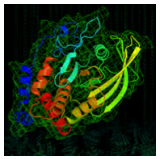
BERKELEY LAB
Lawrence Berkeley National Laboratory

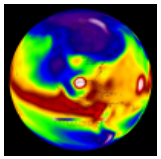
Welcome!

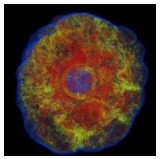
Computational Research and Theory Facility (CRT)

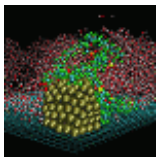












NERSC 40 YEARS at the FOREFRONT

Jeff Broughton

NERSC Deputy for Operations

CRT is complete, and NERSC is moving in!

CRT will help meet the needs of future Exascale system

- **Accommodate system trends**
  - Power and power density is increasing
  - Systems are getting heavier
  - Expect (exotic) liquid cooling
- **Accommodate system growth**
  - Capability for flexible expansion is key
- **Improve energy efficiency**
  - Exploit Bay Area environment for “free cooling”
  - Access to lower-cost WAPA power
- **Enable co-design of next generation systems, networks and applications**
  - Closer interaction with the rest of CS
  - Collaborative CS with UCB

Computational Research and Theory Building (CRT)

- **Four story, 140,000 GSF facility for scientific computing including:**
  - 20,000+9,870 ASF High Performance Computing Floor
  - 41,000 ASF office and conference area; ~300 offices
- **$143M UC Sponsored Building**
  - No long term commitment or decommissioning costs
  - No major capitalization or appropriations costs
- **$19.8M DOE Funded Data Center**
  - Power and cooling expansion for NERSC systems
- **Notable Features**
  - Free cooling
  - Heat recovery
  - Seismically isolated floor
  - 400Gb/s link to Oakland

Power and Cooling Capacity

	Move-in	Drop-in Expansion Capability
Power feeders	27MW redundant 42MW non-redundant	Same
Power Substations	5 substations @ 2.5 MW	11 = 27.5 MW
UPS	1.0+0.5 MW	2.0+1.0 MW
Generator	1 @ 1.25 MW	2 = 2.5 MW
AHUs	3+1 redundant @ 60K CFM / 0.5MW = 1.5 MW	30 = 15 MW
Cooling Towers	3+1 @ 3.375MW = 10.25 MW	6+1 = 20.25 MW
Chillers	None	2 x 550 ton

Plan for Major Systems

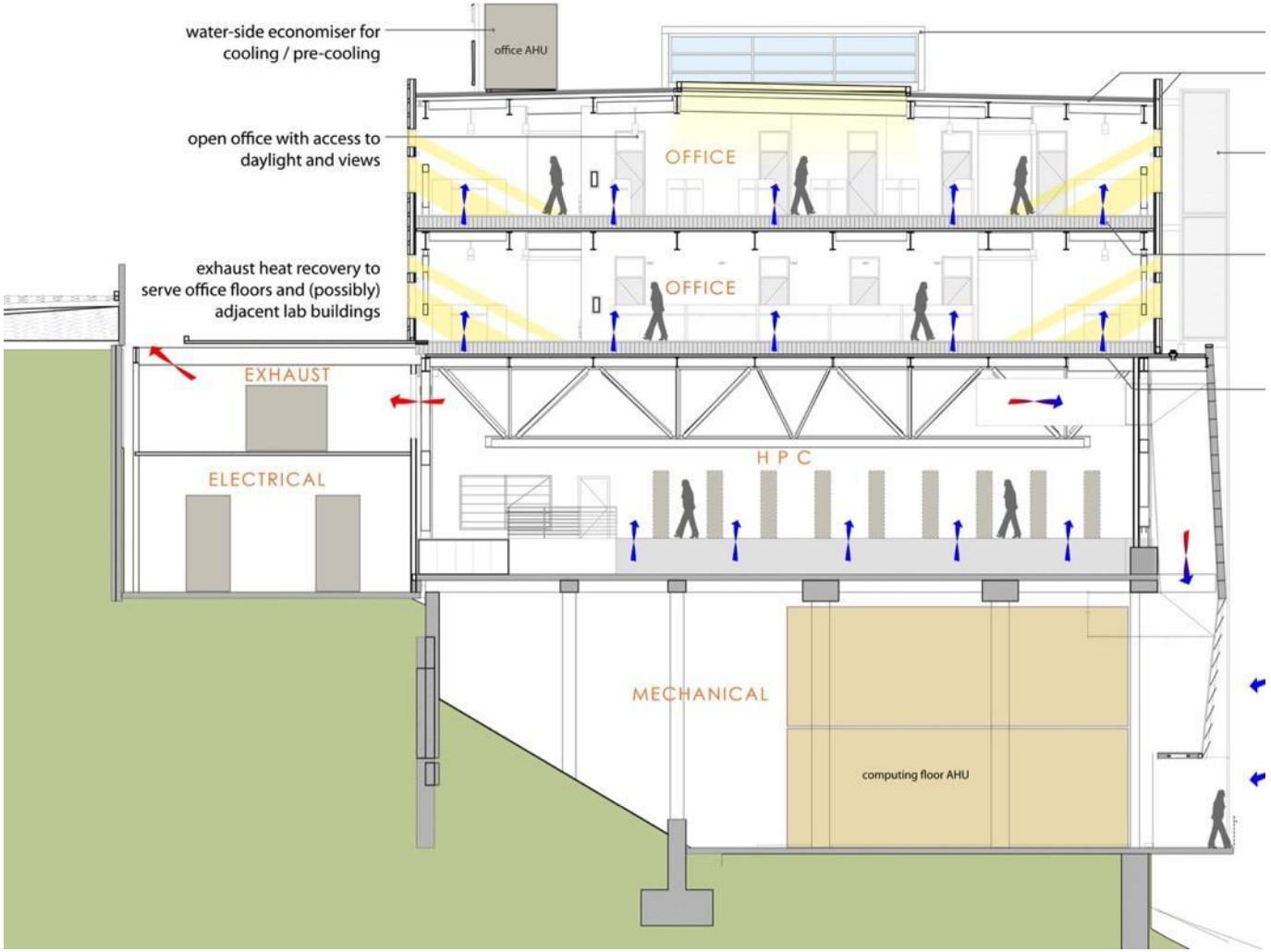
Systems	Plan	Target Date
NERSC-6 (Hopper)	Retire at OSF	Dec 2015
NERSC-7 (Edison)	Move from OSF to CRT	Dec 2015
NERSC-8 (Cori Phase 1)	New install at CRT	Aug 2015
NERSC-8 (Cori Phase 2)	New install at CRT	Summer/Fall 2016
Carver	Retire at OSF	Sept 2015
Mendel (JGI)/PDSF (Physics)	Move from OSF to CRT	Feb 2016
Mendel+	New install at CRT	July 2015
Global File System	Move physical replica and sync over network	Oct 2015 – Feb 2016 Various dates for individual file systems
HPSS Tape Archive	Remain at OSF until we need to vacate	Sept 2016?

“The coldest winter I ever spent was a summer in San Francisco.”

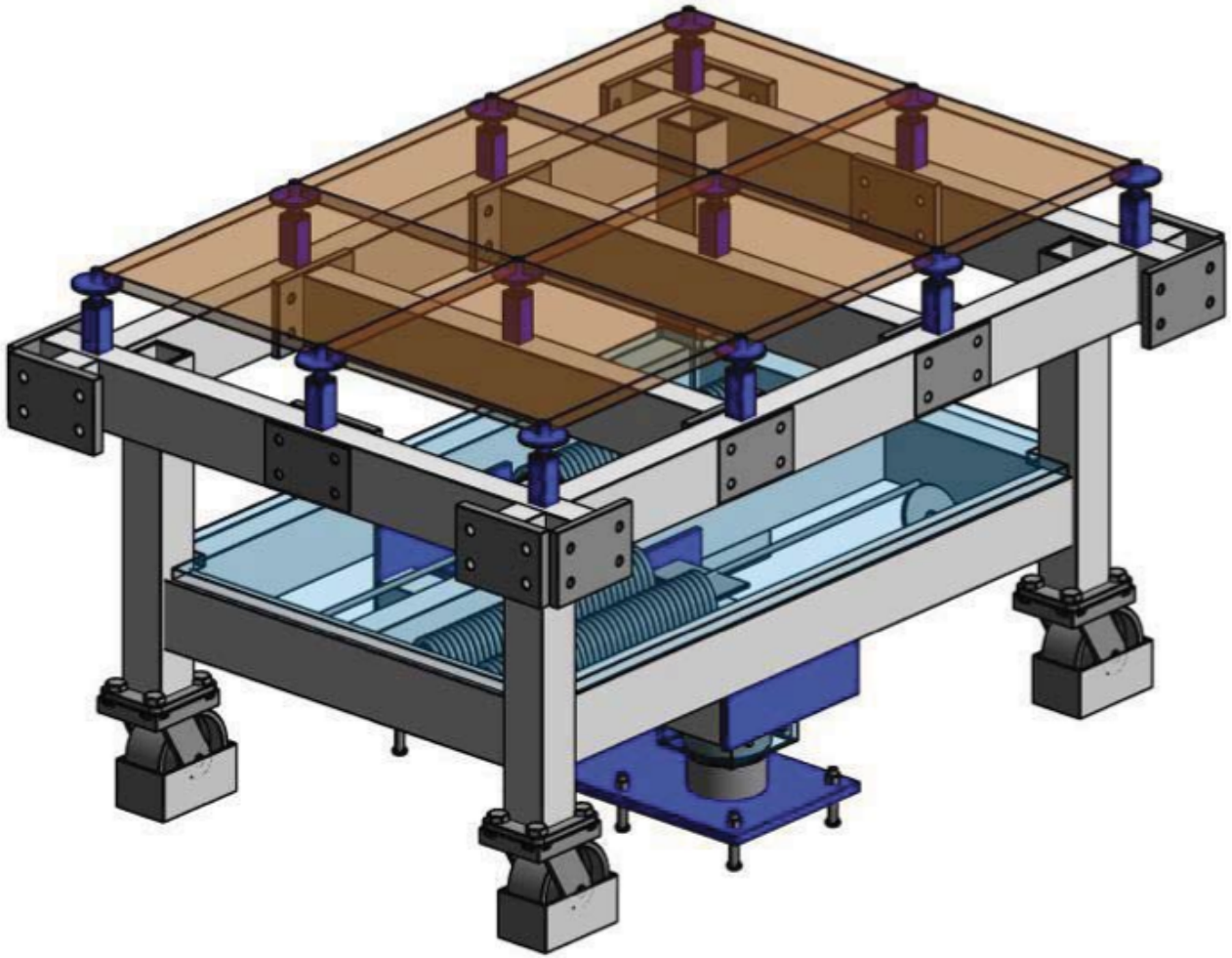
Free cooling provides exceptional energy efficiency

- **LBNL's location and the CRT design enables**
  - Power Usage Effectiveness (PUE): <1.1
  - Data Center infrastructure Efficiency (DCiE): >0.91
- **Air cooling**
  - 75°F air year round without chillers
- **Liquid cooling**
  - 74°F water year round without chillers
- **Computer room exhaust heat used to heat office floors**
- **Save ~50% per year on power costs**
  - Free cooling + WAPA power

Building Cross Section

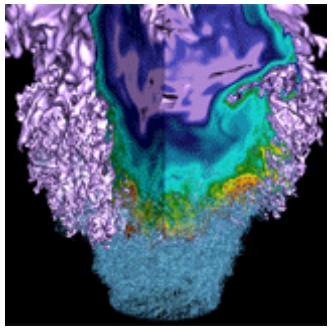


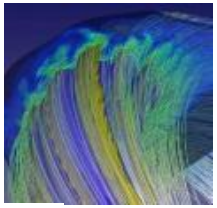
Seismic floor isolates systems from severe earthquakes

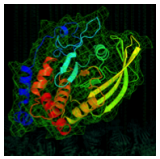


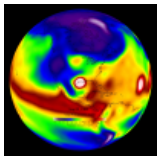
Raised floor rides on a rolling substructure with a spring dampening mechanism

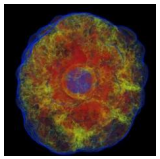
NERSC Move: Outage Schedule and User Impact

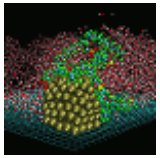












Helen He

NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Carver Has Retired

- **Retired on Sept 30, 2015 at noon**
- **Users can access all Carver file systems mounted on other NERSC systems**
  - Notice no more access to /global/scratch2 after Oct 14.
- **We are here to help migrating your workflow to Edison if needed.**
  - http://www.nersc.gov/users/computational-systems/carver/retirement-plans/

Global Scratch Retirement

- **/global/scratch2, usually referred to as $GSCRATCH has become read-only on Sept 30, 2015**
  - Some confusions last week: write access still available from certain login/compute nodes on Hopper/Edison.
  - Current situation: either unmounted or mounted as read-only except on a few Edison login nodes.
  - Bottom line: please do not write to it even if you can.
- **Will *retire* on next Wed, Oct 14, 2015 at noon.**
  - No user access after this time!
- **Please backup your important files to HPSS or another permanent file storage before Oct 14.**
  - Do not wait until the last day. Backup may take longer than expected.
- **Contact consult@nersc.gov for any questions**

Main Timeline

Event	Date
Global Scratch retires	Oct 14, 2015
Cori Phase 1 available to all users	Nov 2015 (estimate)
Edison offline to move to CRT	Dec 2015 (estimate to last 6 weeks), back online in Jan 2016 with SLURM batch scheduler
Hopper retires	Dec 15, 2015 (after Cori Phase 1 is stable)
Global Home replicate/install at CRT; Global Project replicate at CRT	Dec 2015
JGI and PDSF file systems move to CRT	Feb 2016
Mendel moves to CRT	Feb 2016, affects JGI, PDSF, Materials Project and Babbage.

More Details (no change from last NUG)

- **Dec 2015**
  - Global homes file system replicated and installed at CRT. Reduced bandwidth. No outage.
  - Global project file systems replicated at CRT. Up to 5 days of /project outage.
- **Feb 2016**
  - /global/projectb, /global/dna, /global/seqfs replicated at CRT. Reduced bandwidth. Up to 7 days outage of these JGI file systems.
  - Some PDSF file systems relocated to CRT. ~2 weeks outage of these PDSF file systems.

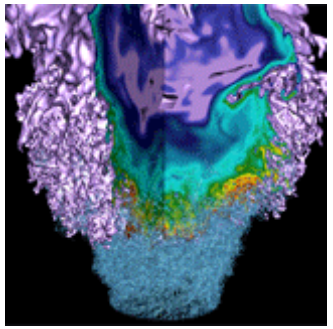
More Details (no change from last NUG)

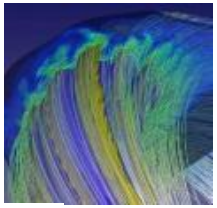
- **Feb 2016**
  - A cluster (internal name “Mendel”) providing resources to PDSF, JGI, the Materials Project, and Babbage will move to CRT.
  - Total amount of resources for PDSF and JGI will be reduced. Not an outage since new hardware already in place at CRT.
  - Materials Project: ~ 3 weeks outage
  - Babbage: up to 1 month outage

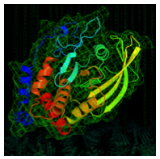
Big Message to Users

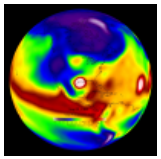
- **Please use your allocation now instead of end-of-year crunch.**
  - During Edison's move to CRT (~6 weeks), Cori will be the only MPP system available to users.
- **During the move, file system resources will be spread across OSF and CRT. Available I/O bandwidth to global file systems will be impacted.**
- **HPSS will remain at OSF until other moves complete**
- **Follow up with move updates at:**
  - https://www.nersc.gov/users/move-to-crt/

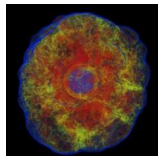
Cori Phase 1 Update

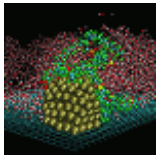










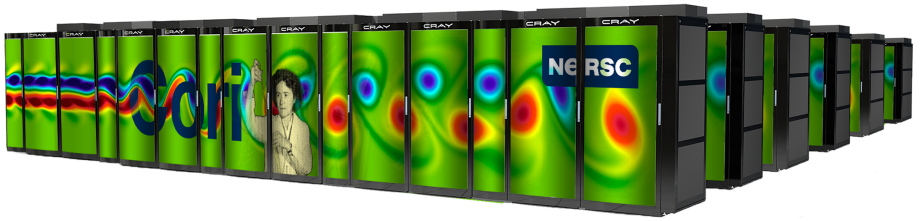


Wahid Bhimji

NERSC Users Group
October 8th 2015

Cori Overview (reminder)

- **Phase 2 (coming mid-2016) - over 9,300 '[Knights Landing](#)' compute nodes**

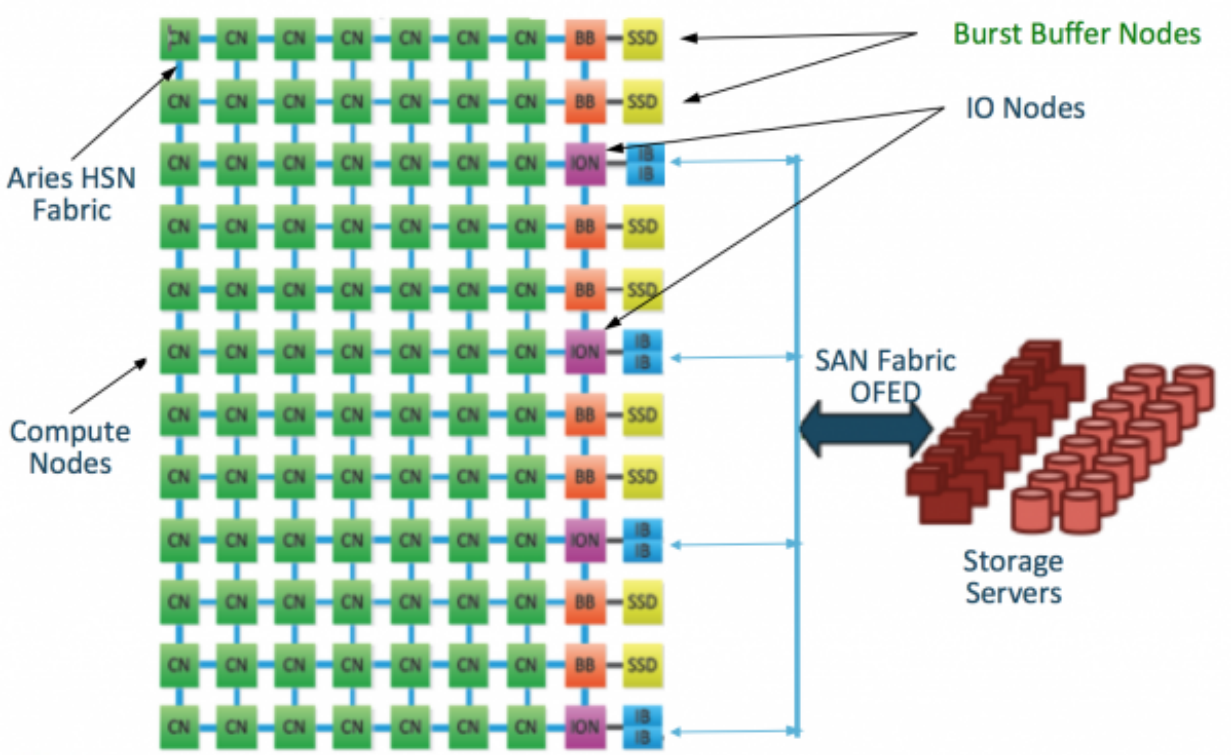


Phase 1 (being installed now):

- **1630 Compute Nodes**
- Two Haswell processors/node,
  - 16 cores/processor at 2.3 GHz
  - 128 GB DDR4 2133 Mhz memory/ node(some 512 /768 GB)
  - Cray Aries high-speed "dragonfly" topology interconnect
  - 12+ login nodes for advanced workflows and analytics
  - SLURM batch system
- **Lustre File system (also installed now)**
  - 28 PB capacity, >700 GB/sec peak performance
  - Ultimately mounted to other systems

Burst Buffer (reminder)

https://www.nersc.gov/users/computational-systems/cori/burst-buffer/



- ~1.5PB capacity, ~1.5TB/s for full Cori System
- Half with Phase 1 - being installed over the coming weeks
- Available via SLURM batch system integration with Cray 'Data Warp' Software

- **IO improvements: high bandwidth** reads and writes, e.g. checkpoint/restart; **high IOP/s** (input-output operations per second), e.g. non-sequential table lookup; **out-of-core** applications
- **Workflow performance improvements: coupling applications,** using the BB as interim storage; Optimizing node usage by changing node concurrency part way through a workflow (using a persistent BB reservation)
- **Analysis and Visualization: In-situ / in-transit; Interactive** (using a persistent BB reservation)

Burst Buffer Early Users

https://www.nersc.gov/users/accounts/allocations/burst-buffer-early-user-program/

https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/2015/early-users-to-test-new-burst-buffer-on-cori/

- Lots of projects applied to be early users of the Burst Buffer - and help us configure it!
  - Spanning range of science and burst buffer use cases
- Some chosen for NERSC support (below) - many others enabled for early use

NERSC-supported: New Efforts

- Nyx/BoxLib cosmology simulations, Ann Almgren, Berkeley Lab (HEP)
- Phoenix: 3D atmosphere simulator for supernovae, Eddie Baron, University of Oklahoma (HEP)
- Chombo-Crunch + VisIt for carbon sequestration, David Trebotich, Berkeley Lab (BES)
- Sigma/UniFam/Sipros bioinformatics codes, Chongle Pan, Oak Ridge National Laboratory (BER)
- XGC1 for plasma simulation, Scott Klasky, Oak Ridge National Laboratory (FES)
- PSANA for LCLS, Amadeo Perazzo, SLAC (BES/BER)

NERSC-supported: Existing Engagements

- ALICE data analysis, Jeff Porter, Berkeley Lab (NP)
- Tractor: Cosmological data analysis (DESI), Peter Nugent, Berkeley Lab (HEP)
- VPIC-IO performance, Suren Byna, Berkeley Lab (HEP/ASCR)
- YODA: Geant4 sims for ATLAS detector, Vakhtang Tsulaia, Berkeley Lab (HEP)
- Advanced Light Source SPOT Suite, Craig Tull, Berkeley Lab (BES/BER)
- TomoPy for ALS image reconstruction, Craig Tull, Berkeley Lab (BES/BER)
- kitware: VPIC/Catalyst/ParaView, Berk Geveci, kitware (ASCR)

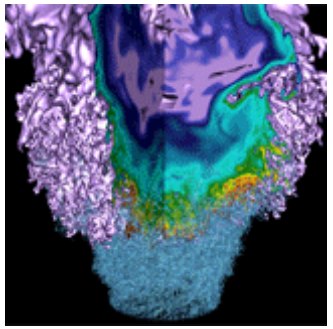
Coming very soon!

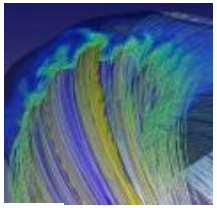
- Currently installed and being configured
- Limited user access from late-October alongside machine acceptance
- On target to get Cori Phase 1 into production mid-Dec before Hopper retirement

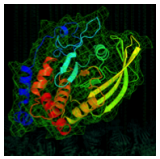
Full configuration details and user guides (including SLURM transition and Burst Buffer guide) becoming available at:

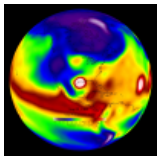
- https://www.nersc.gov/users/computational-systems/cori/

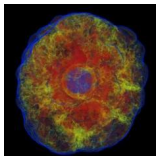
Edison Update: no news!

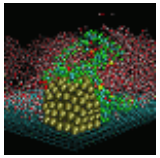






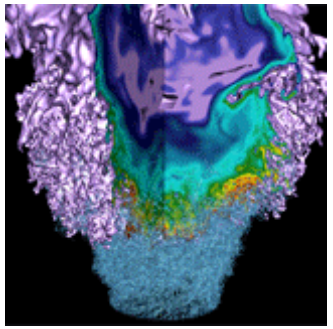


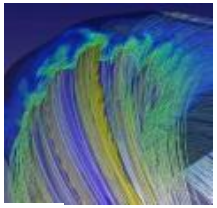


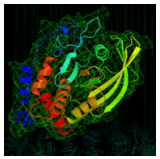


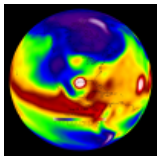
Zhengji Zhao

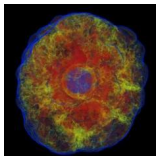
NESAP Update

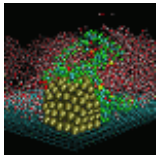












NERSC 40 YEARS at the FOREFRONT

Jack Deslippe

NESAP Highlights

- Fourth Dungeon Session in Portland Complete. EMGEO, WARP, VASP. Next Opportunity in January.
- 5 Day IXPUG workshop held at CRT. 150+ Attendees. DFT + Particle Accelerator dev. workshops.
- 3 Postdocs arrived at NERSC. 3 More on the way.

Taylor Barnes
Quantum ESPRESSO

Brian Friesen
Boxlib

Andrey Ovsyannikov
Chombo-Crunch

- New documentation: https://www.nersc.gov/users/computational-systems/cori/application-porting-and-performance/

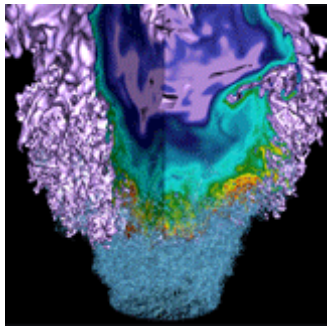
Test HBM Directives on Edison

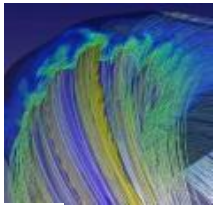
On Edison (NERSC Cray XC30):

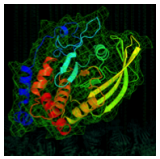
```
real, allocatable :: a(:,:), b(:,:), c(:)
!DIR$ ATTRIBUTE FASTMEM :: a, b, c

% module load memkind jemalloc
% ftn -dynamic -g -O3 -openmp mycode.f90
% export MEMKIND_HBW_NODES=0
% aprun -n 1 -cc numa_node numactl --membind=1 --cpunodebind=0 ./myexecutable
```

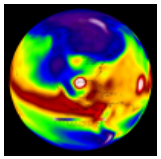
More info: https://www.nersc.gov/users/computational-systems/cori/application-porting-and-performance/using-high-performance-libraries-and-tools/

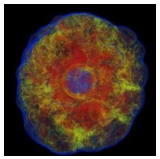
IXPUG Summary

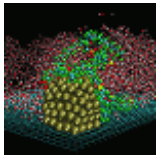












Richard Gerber &
Jack Deslippe

NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

IXPUG 2015

Intel Xeon Phi Users Group: An independent performance-focused user group

4+ days of technical info related to application readiness for Phi (KNL, Cori)

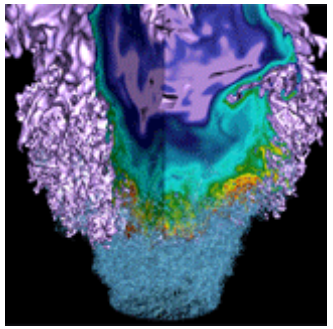
Held in NERSC's new building at Berkeley Lab

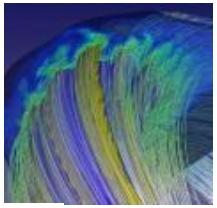
120+ local attendees
30-60 remote
(Had to close registration)

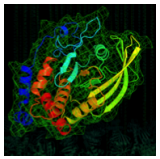
Held in NERSC's new building at Berkeley Lab

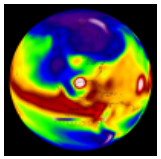
Presentations: ixpug.org

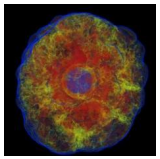
Science Byte

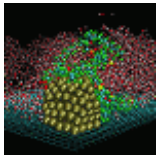












NERSC 40 YEARS at the FOREFRONT

Debbie Bard

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Science Byte

Spiraling laser pulses could change nature of graphene

- Graphene is a pretty cool material.
- But it doesn't do everything: scientists have been trying to turn graphene into a semiconductor (essential component of microelectronics).
  - Want to turn graphene from a metal (where electron flow freely) to an insulator.
- Preliminary work suggested a band gap could be produced using circularly polarised light...

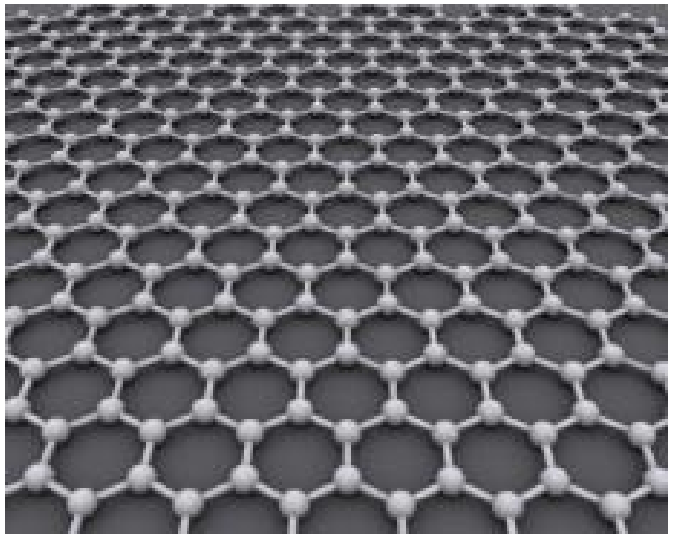


Illustration of the honeycomb structure of graphene.

Sentef, Claassen, Kemper et al., 2015, Nature Communications **6**, 7047.

Science Byte

Spiraling laser pulses could change nature of graphene

- Hopper was used to simulate what would happen when hit by circularly polarised light, in a *realistic* experimental setup.
- Encode information - like bit flipping.
  - potential applications beyond graphene.
  - low-energy electronics, quantum computing, light detectors...
- ***can create and control new form of matter with light***

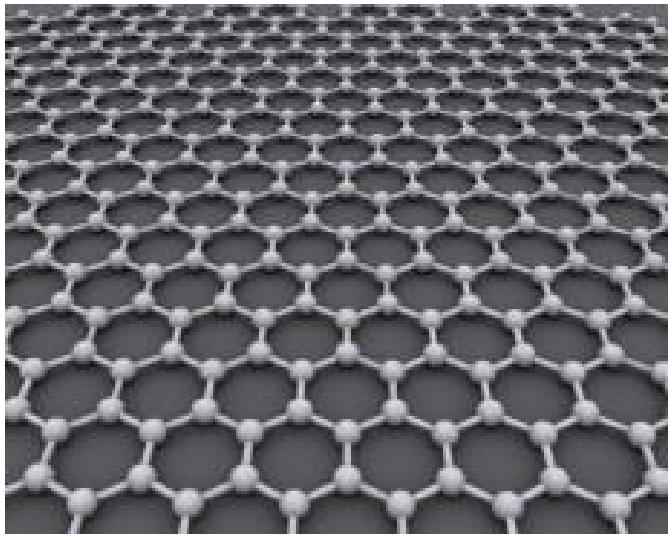


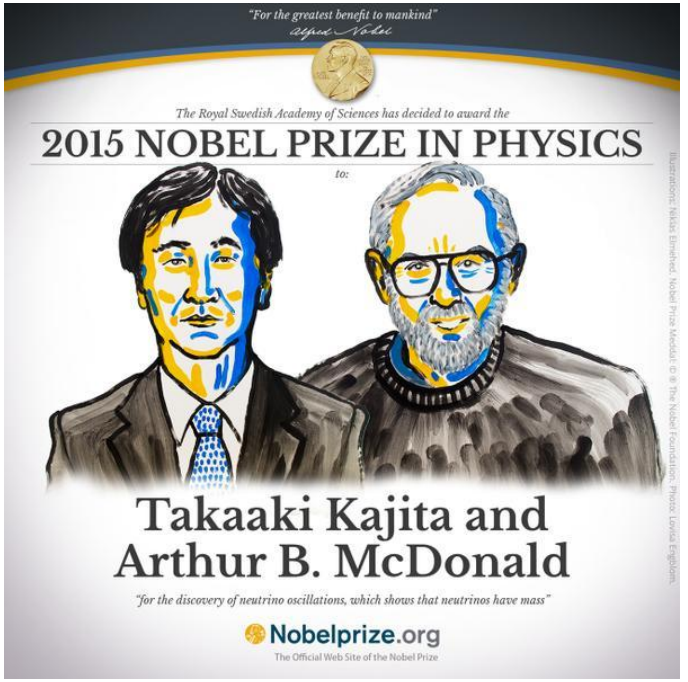
Illustration of the honeycomb structure of graphene.

Sentef, Claassen, Kemper et al., 2015, Nature Communications **6**, 7047.

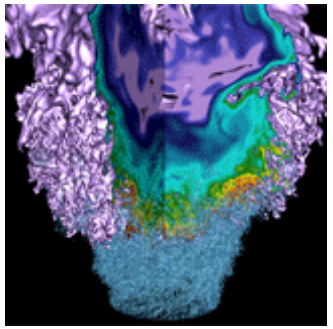
2015 Nobel Prize in Physics

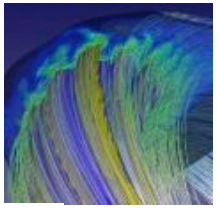
"For the discovery of neutrino oscillations, which shows that neutrinos have mass"

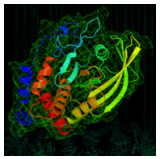
- Work done by Arthur B. McDonald's group with Sudbury Neutrino Observatory (SNO) was supported by data analysis and archiving at NERSC.
- We're thrilled to be associated with such exciting science!

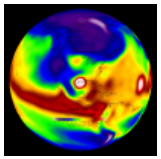


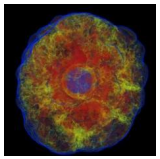
Application Portability Best Practices Workshop Highlights

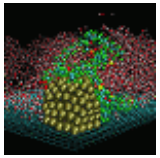












Katie Antypas

NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Background

- **There will be at least two different architectures in DOE ASCR supercomputers in the the next 5 years**
  - NERSC and ALCF will deploy Cray-Intel Xeon Phi manycore based systems in 2016 and 2018
  - OLCF will deploy and IBM Power/NVIDIA based system in 2017
- **Question: Are there best practices for achieving performance portability across architectures?**
- **Workshop held in Bethesda MD Sept. 15-17th with about 100 participants**

Charge questions

- What are the practices used by scientific application codes today to achieve portability?
- What are the practices that scientific software code can use to increase portability across architectural differences? What is the impact to performance?
- Are any of these practices identifiable as 'best practices'?
- How can the ASCR, and ASCR and NNSA computing facilities support these efforts?
- What new research needs to be carried out to support continued best practices?

But wait… what is the definition of application performance portability?

- **We came up with 3 definitions:**
  - "user portability" -- the science user does not see a difference in performance or answer -- loosest definition because it can include 2 branches of the code
  - "library portability" -- the application developer does not see a difference (or sees only a small difference?) in using the library
  - "code portability" -- no lines of code change between platforms and performance is preserved

3 Breakout sessions

- **Application Architecture**
- **Libraries and Tools**
- **Software Engineering**

- **I co-lead Application Architecture and will share some *preliminary* findings**

Practices – Best and Otherwise

- **Applications really truly running on both GPU and CPU architectures today create a code structure that allows a 'plug-and-play' of architecture specific modules**
  - No applications currently use the exact code base and run with performance across GPU and CPU systems
- **Use libraries that have been optimized for a particular architecture when possible**
- **Abstract memory management in code so it can easily be replaced**
- **Write modular, layered code without vendor specific compiler optimizations**
- **Pay careful attention to data structure layout(s) which improves data locality and can improve portability on many architectures**

Preliminary

Emerging or Niche practices

- Use frameworks like RAJA, KOKKOS or TIDA to enable portability
- Consider using domain specific languages
- Use LLVM as a backend code generator/compiler/translator
- Use autotuning to find best parameters for different architectures
- Architecting code in a tasking model

Preliminary

Failures

- Trying to completely re-write a code while managing an active research and code development
- Relying on libraries that then lose funding
- Neglecting code maintenance
- Developing performance improvements/tuning separate from science teams – which never gets back into main production code repository

Preliminary

Opportunities

- Train early career staff on multi-disciplinary computational science areas (performance, applications, architecture and algorithms)
- Use libraries to encapsulate architecture specific optimizations
- Use OpenMP 4.x as a standard that is portable across architectures
- Adopt OpenACC features into OpenMP and focus investments into one standard
- Use DSLs, compilers, translators and code generators to achieve architecture tuned performance

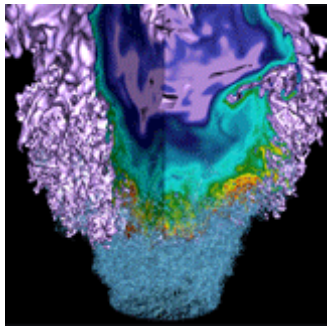
Preliminary

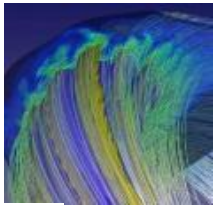
Risks

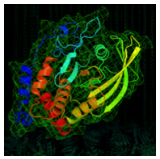
- Runtime/compiler technologies will not keep up limiting usability of higher level frameworks and new parallelism features of languages
- Over templating and over use of abstractions will lead to long compile times and clunky code
- Legacy code base will lead to make conservative choices
- Reliance on external libraries that are not yet optimized for new architectures
- There exists no portable API for expressing data movement between memory hierarchies
- Directive based programming models do not address day layout/data manipulation capabilities

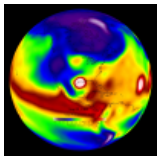
Preliminary

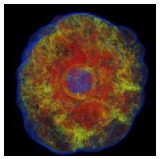
Jobs @ NERSC

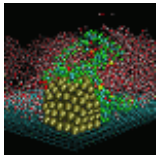












NERSC 40 YEARS at the FOREFRONT

Debbie Bard

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

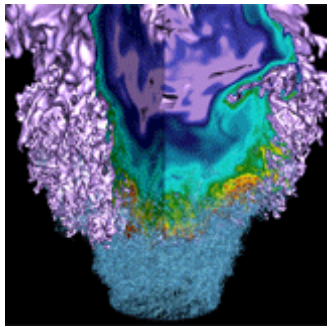
NERSC is hiring!

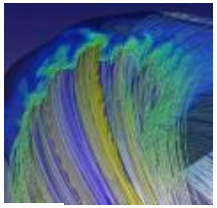
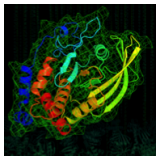
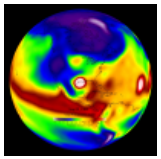
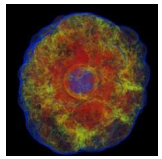
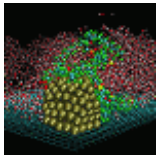
- **We have positions open at several levels at NERSC**
- For example,
  - High Performance Computing Consultant with the Users Services Group
  - Advanced Technologies Group Engineer
  - High Performance Data Analytics Engineer
  - HPC Systems Engineer
  - Computational Systems Engineer

NERSC Users make the best employees!

http://cs.lbl.gov/careers/careers-and-fellowships/

Mini-Tutorial: Nested OpenMP



Helen He

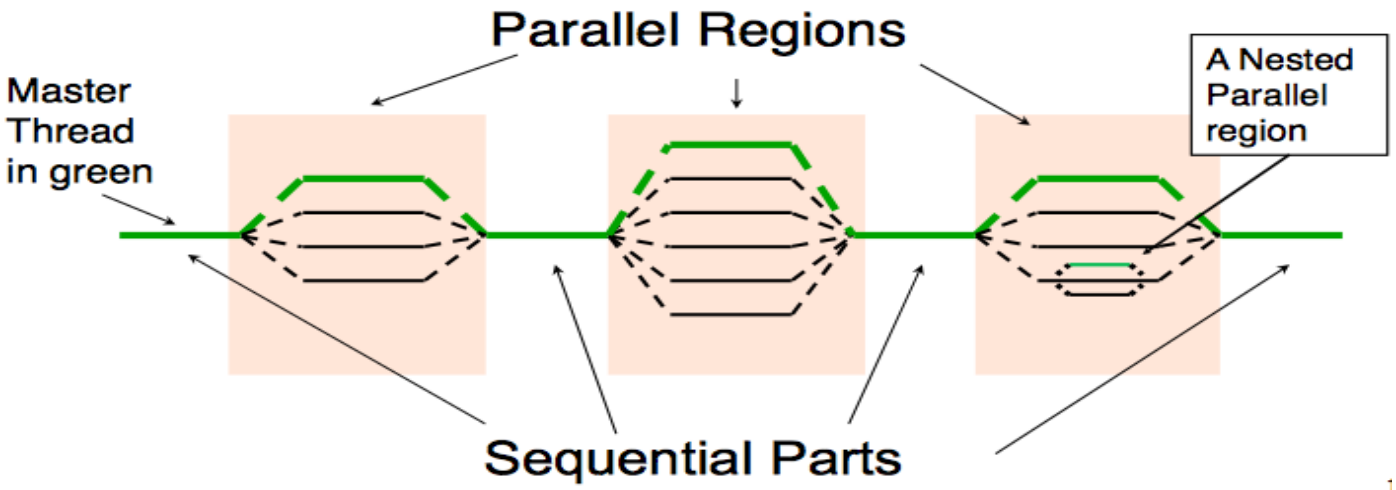
NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

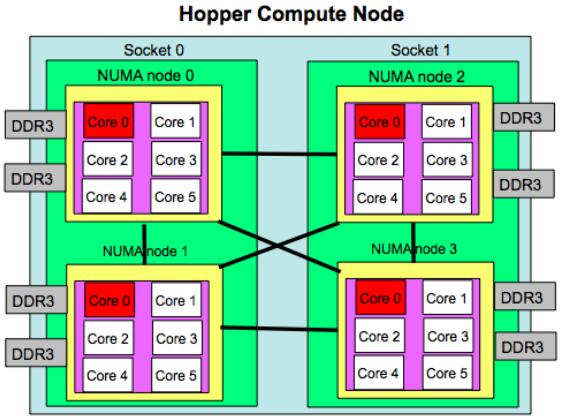
BERKELEY LAB

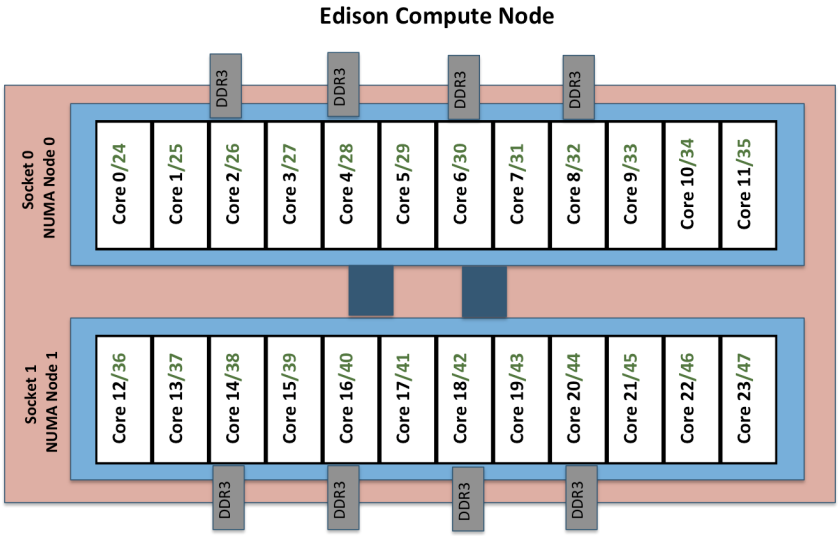
OpenMP Execution Model

- **Fork and Join Model**
  - Master thread forks new threads at the beginning of parallel regions.
  - Multiple threads share work in parallel.
  - Threads join at the end of the parallel regions.



Hopper/Edison Compute Nodes



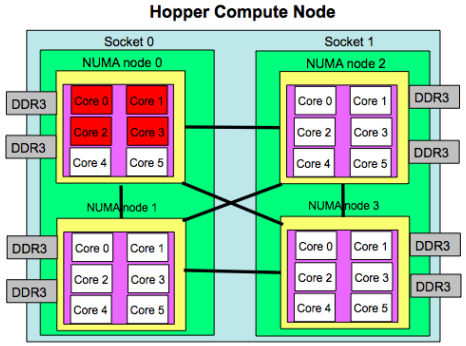


- Hopper: NERSC Cray XE6, 6,384 nodes, 153,126 cores.
  - 4 NUMA domains per node, 6 cores per NUMA domain.
- Edison: NERSC Cray XC30, 5,576 nodes, 133,824 cores.
  - 2 NUMA domains per node, 12 cores per NUMA domain. 2 hardware threads per core.
- Memory bandwidth is non-homogeneous among NUMA domains.

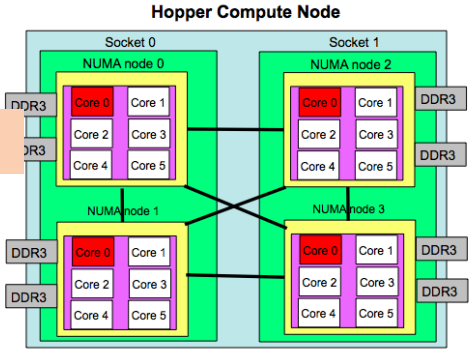
MPI Process Affinity: aprun "-S" Option

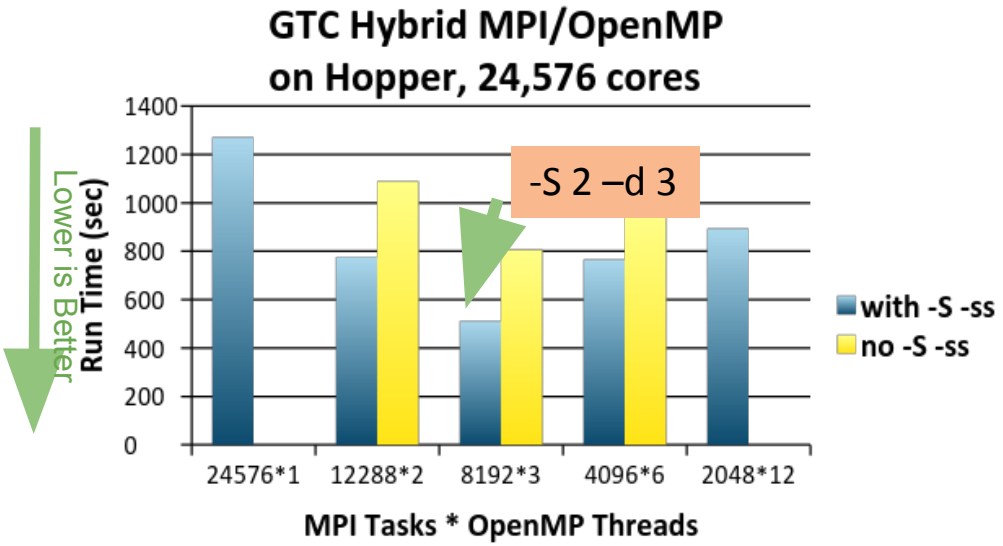
- **Process affinity: or CPU pinning, binds MPI process to a CPU or a ranges of CPUs on the node.**
- **Important to spread MPI ranks evenly onto different NUMA nodes.**
- **Use the "-S" option for Hopper/Edison.**

aprun –n 4 –d 6



aprun –n 4 –S 1–d 6



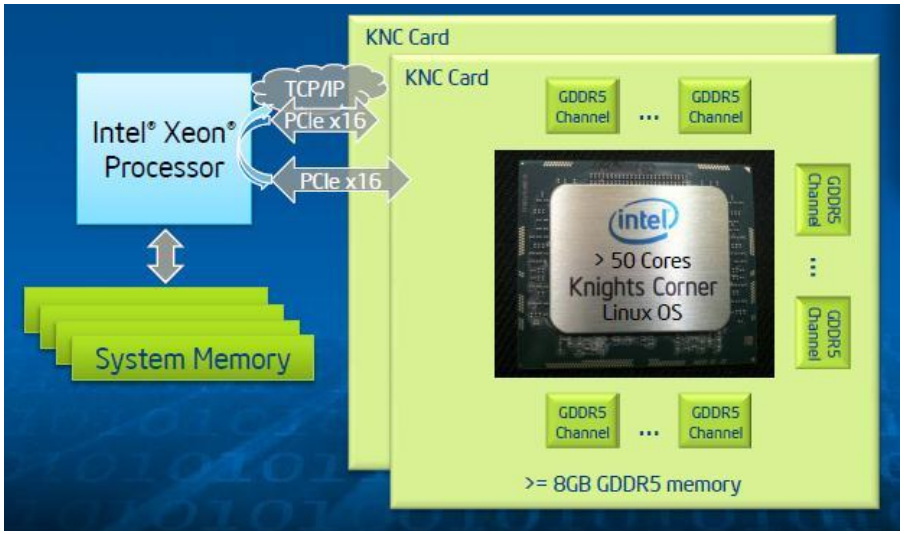


Thread Affinity: aprun "-cc" Option

- **Thread affinity: forces each process or thread to run on a specific subset of processors, to take advantage of local process state.**
- **Thread locality is important since it impacts both memory and intra-node performance.**
- **On Hopper/Edison:**
  - The default option is "-cc cpu" (use it for non-Intel compilers), binds each PE to a CPU within the assigned NUMA node.
  - Pay attention to Intel compiler, which uses an extra thread.
    - Use "-cc none" if 1 MPI process per node
    - Use "-cc numa_node" (Hopper) or "-cc depth" (Edison) if multiple MPI processes per node

NERSC KNC Testbed: Babbage

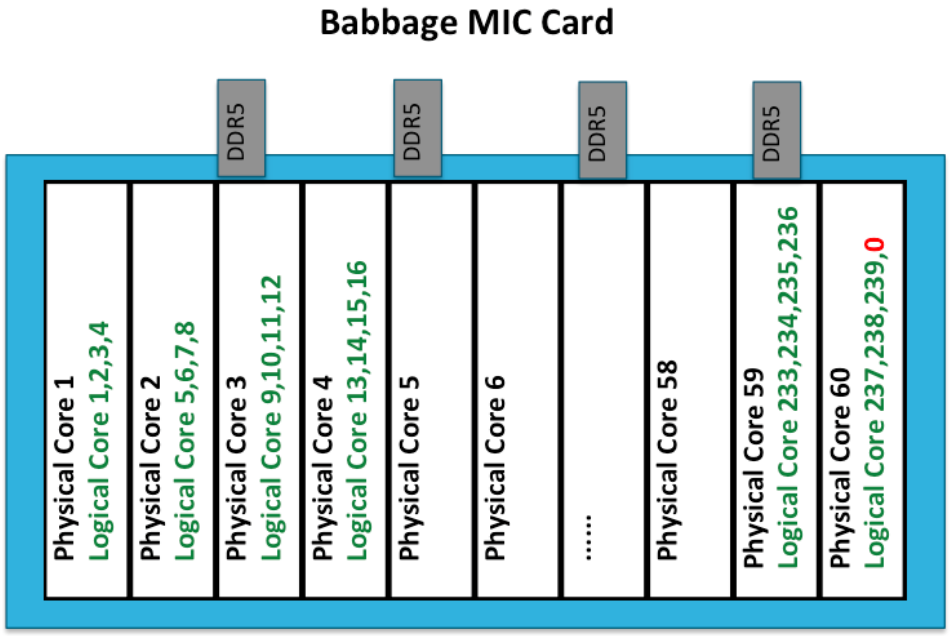
- **NERSC Intel Xeon Phi Knights Corner (KNC) testbed**
- **45 compute nodes, each has:**
  - Host node: 2 Intel Xeon Sandybridge processors, 8 cores each.
  - 2 MIC cards each has 60 native cores and 4 hardware threads per core.
  - MIC cards attached to host nodes via PCI-express.
  - 8 GB memory on each MIC card
- **Recommend to use at least 2 threads per core to hide latency of in-order execution.**



To best prepare codes on Babbage for Cori:

- Use "native" mode on KNC to mimic KNL, which means ignore the host, just run completely on KNC cards.
- Encourage to explore single node optimization for threading scaling and vectorization on KNC cards with problem sizes that can fit.
- "Symmetric", "Offload" modes on KNC and "OpenMP 4.0 target" work, but are not our promoted usage models for Babbage.

Babbage MIC Card



Babbage: NERSC Intel Xeon Phi testbed, 45 nodes. 2 MIC cards per node.

- **1 NUMA domain per MIC card: 60 physical cores, 240 logical cores. OpenMP threading potential to 240-way.**
- **KMP_AFFINITY, KMP_PLACE_THREADS, OMP_PLACES, OMP_PROC_BIND for thread affinity control**
- **I_MPI_PIN_DOMAIN for MPI/OpenMP process and thread affinity control.**

Full OpenMP 4.0 Support in Compilers

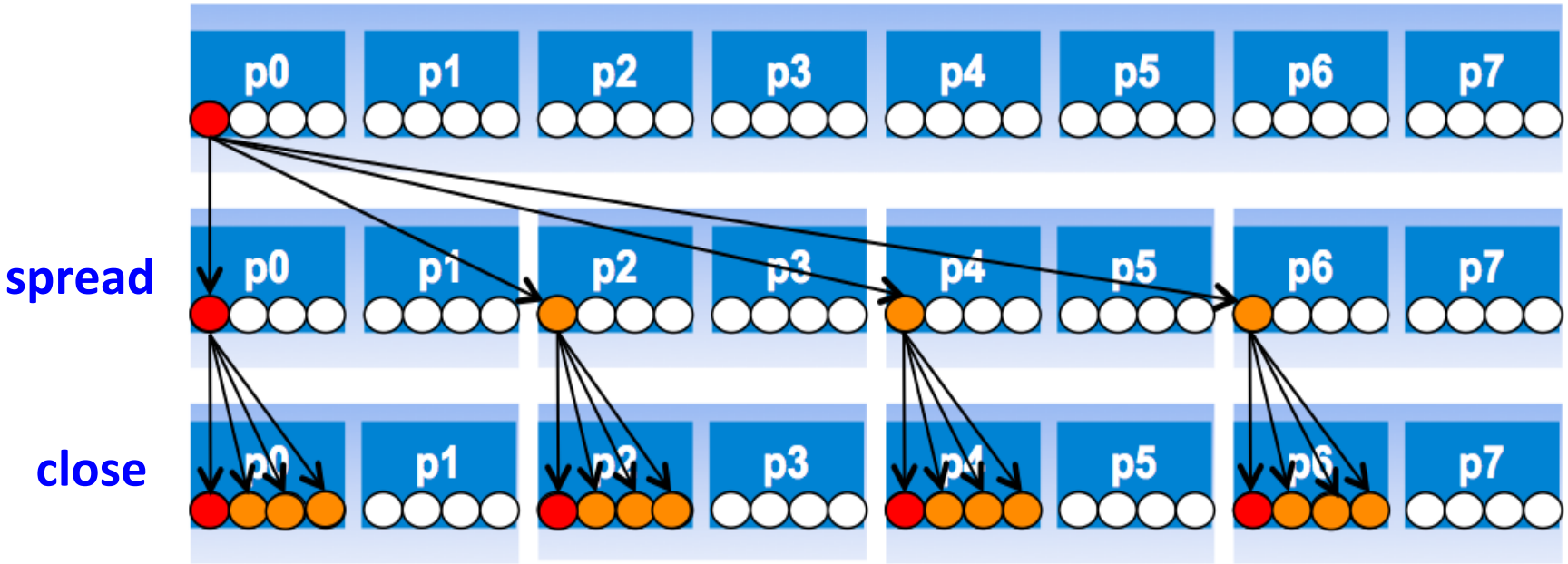
- **GNU compiler**
  - From 4.9.0 for C/C++
  - From gcc/4.9.1 for Fortran
- **Intel compiler**
  - From intel/15.0: supports most features in OpenMP 4.0; From Intel/16.0: full support
- **Cray compiler**
  - From cce/8.4.0

Thread Affinity Control in OpenMP 4.0

- **OMP_PLACES: a list of places that threads can be pinned on**
  - threads: Each place corresponds to a single hardware thread on the target machine.
  - cores: Each place corresponds to a single core (having one or more hardware threads) on the target machine.
  - sockets: Each place corresponds to a single socket (consisting of one or more cores) on the target machine.
  - A list with explicit place values: such as:
    - "{0,1,2,3},{4,5,6,7},{8,9,10,11},{12,13,14,15}"
    - "{0:4},{4:4},{8:4},{12:4}"
- **OMP_PROC_BIND**
  - spread: Bind threads as evenly distributed (spreaded) as possible
  - close: Bind threads close to the master thread
  - master: Bind threads the same place as the master thread

Nested OpenMP Thread Affinity Illustration

setenv OMP_PLACES threads
setenv OMP_NUM_THREADS 4,4
setenv OMP_PROC_BIND spread,close



Sample Nested OpenMP Code

```
#include <omp.h>
#include <stdio.h>
void report_num_threads(int level)
{
   #pragma omp single {
      printf("Level %d: number of threads in the
team: %d\n", level, omp_get_num_threads());
      }
}
int main()
{
   omp_set_dynamic(0);
   #pragma omp parallel num_threads(2) {
      report_num_threads(1);
      #pragma omp parallel num_threads(2) {
         report_num_threads(2);
         #pragma omp parallel num_threads(2) {
            report_num_threads(3);
         }
      }
   }
   return(0);
}
```

```
% a.out
Level 1: number of threads in the team: 2
Level 2: number of threads in the team: 1
Level 3: number of threads in the team: 1
Level 2: number of threads in the team: 1
Level 3: number of threads in the team: 1
```

```
% setenv OMP_NESTED TRUE
% a.out
Level 1: number of threads in the team: 2
Level 2: number of threads in the team: 2
Level 2: number of threads in the team: 2
Level 3: number of threads in the team: 2
Level 3: number of threads in the team: 2
Level 3: number of threads in the team: 2
Level 3: number of threads in the team: 2
```

```
Level 0: P0
Level 1: P0 P1
Level 2: P0 P2; P1 P3
Level 3: P0 P4; P2 P5; P1 P6; P3 P7
```

When to Use Nested OpenMP

- **Some application teams are exploring with nested OpenMP to allow more fine-grained thread parallelism.**
  - MPI/Hybrid not using node fully packed
  - Top level OpenMP loop does not use all available threads
  - Multiple levels of OpenMP loops are not easily collapsed
  - Certain computational intensive kernels could use more threads
  - MKL can use extra cores with nested OpenMP

Process and Thread Affinity in Nested OpenMP

- **Achieving best process and thread affinity is crucial in getting good performance with nested OpenMP, yet it is not straightforward to do so.**
- **A combination of OpenMP environment variables and run time flags are needed for different compilers and different batch schedulers on different systems.**

Example: Use Intel compiler with Torque/Moab on Edison:

```
setenv OMP_NESTED true
setenv  OMP_NUM_THREADS 4,3
setenv OMP_PROC_BIND spread,close
aprun -n 2 -S 1 -d 12 –cc numa_node ./nested.intel.edison
```

Edison: Run Time Environment Variables

- **setenv OMP_NESTED true**
  - Default is false for most compilers
- **setenv OMP_MAX_ACTIVE_LEVELS 2**
  - The default was 1 for CCE prior to cce/8.4.0
- **setenv OMP_NUM_THREADS 4,3**
- **setenv OMP_PROC_BIND spread,close**
- **setenv KMP_HOT_TEAMS 1**
  - Intel only env. Default is false
- **setenv KMP_HOT_TEAMS_MAX_LEVELS 2**
  - Intel only env. Allow nested level OpenMP threads to stay alive instead of being destroyed and created again to reduce thread creation overhead.
- **aprun -n 2 -S 1 -d 12 –cc numa_node ./nested.intel.edison**
  - Use -d for total number of threads (products of num_threads from all levels). –cc numa_node to allow threads migrate within NUMA node to not affected by Intel's extra manager thread.

Babbage: Run Time Environment Variables

- Set I_MPI_PIN_DOMAIN=auto to get basic MPI process affinity
- Do not set KMP_AFFINITY, otherwise OMP_PROC_BIND will be ignored.
- Use OMP_PLACES = threads (default) instead of sockets
- setenv OMP_NESTED true
- setenv OMP_NUM_THREADS 4,3
- setenv OMP_PROC_BIND spread,close
- setenv KMP_HOT_TEAMS 1
- setenv KMP_HOT_TEAMS_MAX_LEVELS 2
- mpirun.mic -n 2 -host bc1109-mic0 ./xthi-nested.mic |sort

XGC1: Nested OpenMP

- **Always make sure to use best thread affinity. Avoid using threads across NUMA domains.**
- **Currently:**

```
export OMP_NUM_THREADS=6,4
export OMP_PROC_BIND=spread,close
export OMP_NESTED=TRUE
export OMP_STACKSIZE=8000000
aprun -n 200 -N 2 -S 1 -j 2 -cc numa_node ./xgca
```

- **Is a bit slower than (work ongoing):**

```
export OMP_NUM_THREADS=24
export OMP_NESTED=TRUE
export OMP_STACKSIZE=8000000
aprun -n 200 -d 24 -N 2 -S 1 -j 2 -cc numa_node ./xgca
```

- **Will try:**

```
export KMP_HOT_TEAMS=1
export KMP_HOT_TEAMS_MAX_LEVELS=2
```

- **Use num_threads clause in source codes to set threads for nested regions. For most other non-nested regions, use OMP_NUM_THREADS env for simplicity and flexibility.**

Courtesy of Robert Hager, PPPL and NESAP XGC1 team.

Use Multiple Threads in MKL

- **By Default, in OpenMP parallel regions, only 1 thread will be used for MKL calls.**
  - MKL_DYNAMICS is true by default
- **Nested OpenMP can be used to enable multiple threads for MKL calls. Treat MKL as a nested inner OpenMP region.**
- **Sample settings**

```
export OMP_NESTED=true
export OMP_PLACES=cores
export OMP_PROC_BIND=close
export OMP_NUM_THREADS=6,4
export MKL_DYNAMICS=false
export KMP_HOT_TEAMS=1
export KMP_HOT_TEAMS_MAX_LEVELS=2
```

NWChem: OpenMP "Reduce" Algorithm

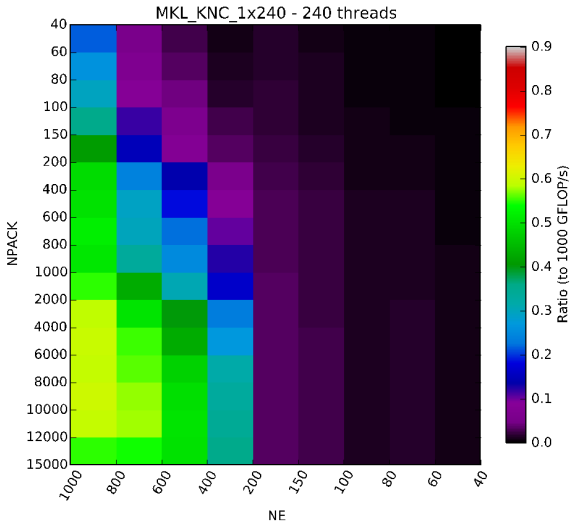
- **Plane wave Lagrange multiplier**
  - Many matrix multiplications of complex numbers, C = A x B
  - Smaller matrix products: FFM, typical size 100 x 10,000 x 100
  - Original threading scaling with MKL not satisfactory
- **OpenMP "Reduce" or "Block" algorithm**
  - Distribute work on A and B along the k dimension
  - A thread puts its contribution in a buffer of size m x n
  - Buffers reduced to produce C
  - OMP teams of threads

Courtesy of Mathias Jacquelin, LBNL

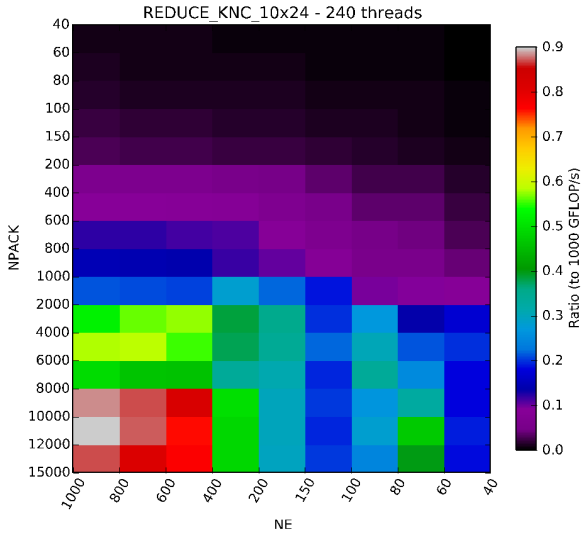
NWChem: OpenMP "Reduce" Algorithm

- Better for smaller inner dimensions, i.e. for FFMs
- Multiple FFMs can be done concurrently in different thread pools
- Threading enables us to use all 240 hardware threads
- Best "Reduce": 10 MPI, 6 teams of 4 threads (nested OpenMP with MKL)

MKL
1MPI, 240 threads

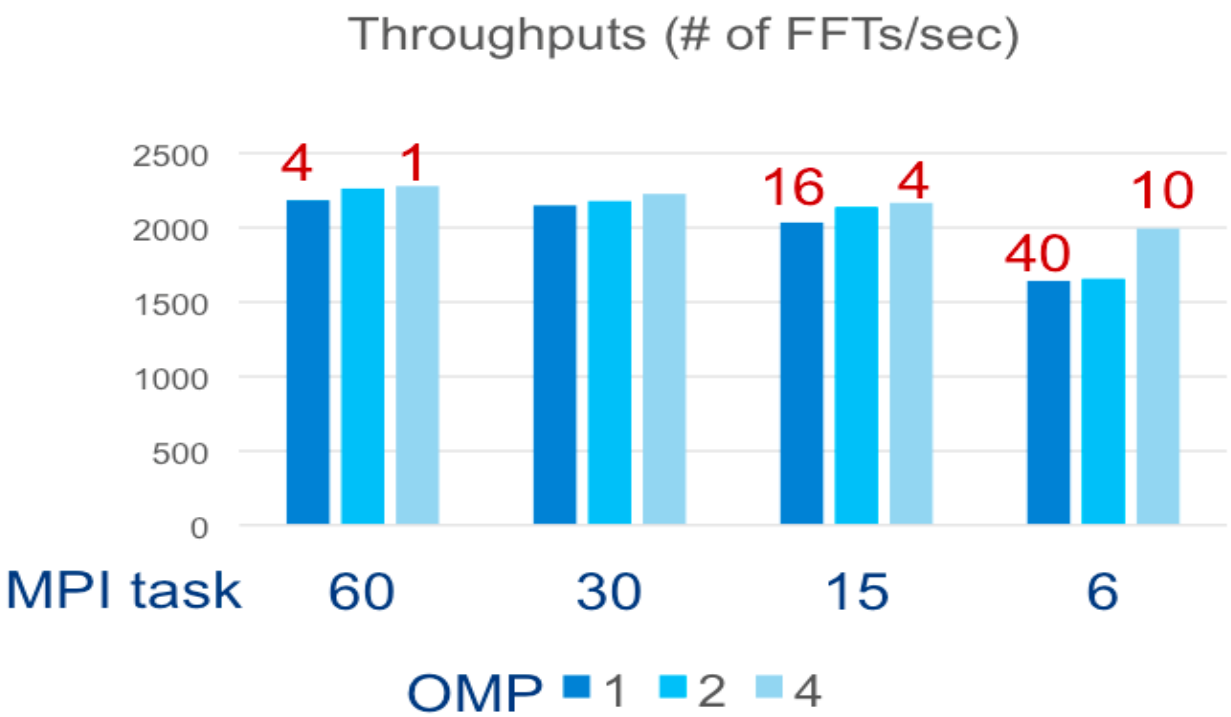


Best "Reduce"
10 MPI, 6 teams of 4 threads



Courtesy of Mathias Jacquelin, LBNL

FFT3D on KNC, Ng=64^3



$$N_{MKL} = 240/(N_{MPI} * OMP)$$

Courtesy of Jeongnim Kim, Intel

Nested OpenMP on NERSC Systems

- **Please see detailed example settings in the "Nested OpenMP" web page:**
  - Run on Edison and Babbage
  - With Intel and Cray compilers
  - Use Torque/Moab and SLURM batch schedulers
  - https://www.nersc.gov/users/computational-systems/edison/running-jobs/using-openmp-with-mpi/nested-openmp/

Thankyou!